

微机屏幕设计技术

孙明珠 主编



天津科技翻译出版公司

津新登字(90)010 号

微机屏幕设计技术

编著 孙明珠

责任编辑 王秀兰

天津科技翻译出版公司出版

(邮政编码:300192)

新华书店天津发行所发行

亚细亚排版公司激光照排

河北省霸州市印刷厂印刷

* * *

开本:787×1092 1/32 印张:8 字数:190 千字

1994 年 3 月第 1 版 1994 年 3 月第 1 次印刷

印数:1—10 000 册

书号: ISBN7-5433-0621-2
TP·10 定价:5.9 元

前 言

随着微机的普及,其应用领域日益广泛。计算机辅助设计(CAD)、计算机辅助教学(CAI)、计算机管理等等,应运而生。这些领域中,在设计程序时都有一个设计屏幕的问题,即:如何设计出美观漂亮的画面,如何设计动画,如何配乐,如何在一幅画面上有多种颜色的文字显示等等。这些问题解决得好,你推出的软件就会令人赏心悦目、倍受欢迎;如果解决得不好,你编出的程序功能即便很好,也会显得逊色。

本书可作为程序设计者的参考书,帮助读者把微机屏幕设计得漂亮生动。作者假定读者对 BASIC 语言已经基本了解,在基础篇中仅对 IBM BASIC 语言作简单总结,重点放在汉字输入、绘图基础、编程技术、彩条的应用、图形文件的建立与调用、动画技术、音响与音乐等基本技能方面;而后在应用篇中介绍人机交互绘图、花型设计、羊毛衫设计、连杆曲线的微机动画、相对轨迹动态显示以及各种课程软件设计等实用程序。通过分析这些程序的设计思想,作者力图提高读者的编程能力,使读者能结合自己的专业知识设计出理想的屏幕文字与图形。

《微机屏幕设计技术》又可作为大学讲义。BASIC 语言简单易学,具有人机对话和较强的作图功能,它是大学中流行最广的计算机语言。一般学生通过中学和大学一年级的学习,基本上都能较好地掌握这门语言,但是如何用之来解决专业问题,还需迈几个较大的台阶。本书的目的就是帮助学生缩小理论到应用之间的距离,跃上一个新的台阶。

本书所涉及的例题及程序均在 IBM PC 微机上调试通过。每节后面均有练习,希望读者能够认真对这些习题编写程序,并上

机运行。这是一门实践性很强的课程，没有必要的实践环节是学不好的。这些例题和习题的答案已经录于一张 5 英寸的“学习盘”中，读者可运行这些程序，以加深对教学内容的理解。但是建议你不要轻易地调用习题的答案，还是独立地作这些练习为好。

天津纺织工学院李兰友教授主审了本书的全部内容，提出了许多宝贵意见，在此表示衷心地感谢。

由于作者水平所限，缺点错误在所难免，敬请批评指正，本人不胜感激。

作者

1994 年 1 月

目 录

基 础 篇

第一节 IBM PC 机 BASIC 语言	1
一、说明及定义语句	1
二、赋值语句	2
三、运算及控制语句	4
四、输出显示语句	7
五、结束语句	9
六、BASIC 编辑命令与文件管理命令	10
练习一	11
第二节 CCDOS 与汉字输入	12
一、CCDOS 简介	12
二、汉字的拼音码输入	14
三、西文字母的汉化	15
四、区位码输入汉字	15
五、联想输入汉字技术	17
六、预选字表的使用	18
七、造字技术	19
练习二	24
第三节 绘图基础与简单动画	24
一、有关概念	24
二、基本图形语句	28
三、应用举例与简单动画	32

练习三	35
第四节 图形文件的建立与调用	36
一、图形数组的建立与存储	36
二、图形文件的调用与再现	37
三、整屏图形的存储与调用	38
练习四	42
第五节 动画技术	43
一、画擦法	43
二、GET——PUT 法	45
三、BLOAD——PUT 法	48
四、应用举例	49
练习五	56
第六节 彩条的使用	56
一、PUT 语句的五种方式	57
二、应用举例	58
练习六	62
第七节 音响与音乐	63
一、音响语句	63
二、音乐语句	63
练习七	73
第八节 编程技术	74
一、“菜单”技术	74
二、链接技术	76
三、拼接技术	77
四、数据的输入与保护	78
五、陷阱技术	81
六、编译 BASIC	84

练习八	86
-----------	----

应 用 篇

第九节 人机交互绘图	87
一、规范图形的绘制	87
二、非规范化图形的绘制	91
三、实用交互绘图程序	95
练习九	109
第十节 织物花型设计	110
一、程序功能	110
二、基础知识	110
三、程序设计	116
练习十	130
第十一节 羊毛衫款式设计	131
一、程序功能	131
二、程序设计	132
练习十一	158
第十二节 连杆曲线的微机动画	159
一、基本理论	159
二、程序设计	160
三、程序的使用	167
练习十二	170
第十三节 相对轨迹动态显示	171
一、理论基础	171
二、程序设计	173
练习十三	177
第十四节 讲授型课件设计	177
一、约束与约束反力	178

二、简谐运动的概念	195
练习十四	201
第十五节 例题型课件设计	201
一、基本思路	202
二、程序设计	204
练习十五	212
第十六节 练习型课件设计	212
一、基本理论	212
二、程序设计	213
练习十六	219
第十七节 计算练习题课件设计	220
一、基本思路	220
二、程序设计	220
练习十七	232
第十八节 实验型课件设计	232
一、测量弹簧——振子的固有频率	233
二、测量悬臂梁的固有频率	239
练习十八	249

基础篇

从第一节到第八节为基础篇，这部分内容是日后编程的基本功，透彻理解这部分内容是非常必要的。在语句格式介绍中，注意三种括号的差别：“〈〉”为必选项，尖括号中的内容必须选择；“[]”为任选项，方括号中的内容可选可不选；“（）”是语句格式中的一部分，须照原样写入。

第一节 IBM PC 机 BASIC 语言

我们假定读者对 BASIC 语言已有初步地了解，这里仅对 IBM PC 高级 BASIC 语言 BASICA 作一简单的回顾与归纳。如读者尚嫌简单，可参阅有关书籍。

我们已经知道，用 BASIC 语言编写的源程序大致可分为开始、赋值、运算、输出、结束五个部分。下面归纳一下各部分中常用的语句。

一、说明及定义语句

此部分的功能是进行必要的说明和运行环境的准备，计有：

* REM _____ 说明语句。它是非执行语句，其后面的内容对程序的运行无影响。“REM”可用单引号“'”代替。

* DIM _____ 数组定维语句。它指出数组的维数及每维所允许使用的最大下标。如：

10 DIM X(20), Y\$(12), A(5,8), B%(600)

语句中，X(20)、Y\$(12)均为一维数组，前者为数值变量，可使用的最大下标为 20，后者为字符串变量，最大下标为 12。A(5,8)为

二维数组，最大下标为(5,8)。B%(600)为整型数组，最大元素编号为 600。

* COMMON _____ 共用变量说明语句，与 CHAIN 语句共同使用，将 COMMON 后面列举的变量传输给下一个程序。

* DEF FN<函数名(变量名表)>=<表达式>_____ 自定义函数语句。变量名表中的变量名为形式参数，在调用该函数时变量名表要给出实在参数。如：

```
10 'PM1-1
20 DEF FNY(X,T)=X^ 2 +5 * T
30 A=1:B=2:X=3:T=4
40 Z= FNY(A,B):PRINT "Z=";Z
50 PRINT "X=";X,"T=";T
60 END
```

运行结果：.

Z= 11

X= 3 T= 4

对照程序与运算结果，可以看出：① 20 句中的 X、T 是形式参数，与程序中的变量 X、T 毫无关系；② 40 句中的 A、B 是实在参数，它们应在调用函数前赋值。

* KEY OFF _____ 关闭功能键显示。

* KEY ON _____ 打开功能键显示。

* CLS _____ 清屏，但不清内存，光标回到屏幕左上角。

* SCREEN n _____ 定义屏幕显示方式(参看第三节)。

二、赋值语句

1. LET 语句

格式：LET <变量名> = <表达式>

说明：

① LET 语句的作用是，将等号右边的表达式的值赋给等号左边的变量。LET 可以略去不写。

② 变量可为数值变量、字符串变量或下标变量。等号两边变量的类型必须一致。

③ 表达式可为单个变量或数字，当表达式为单个字符串时，要用双引号括起来。

2. 键盘输入语句

格式：INPUT [;] ["说明项" (</>)] <变量名 1> [, <变量名 2>...]

说明：

① 说明项前的分号";"为可选项，选用该项时，在键入数据后按回车键，光标不换行。如无该项，则会换行。

② 说明项后选";"时，屏幕显示说明项后紧接着显示问号"?"。选","时，只显示说明项，不显示问号"?"。

③ 变量可为数值变量、字符串变量或下标变量。

④ 输入的数据个数必须与变量名表相对应且类型一致，数据不可为函数或表达式。输入字符串时不必加引号。

⑤ INPUT 语句所列变量全部输入完后方可回车，不能一个一回车。变量之间用逗号相隔。

3. 键盘输入字符串函数

格式 1：A\$ = INKEY\$

功能：读入键盘输入的一个字符，赋值给变量 A\$，且不在屏幕上回显。执行到此句之前，如果键盘上没有任何输入，则 A\$ = ""(两引号之间没有空格)。

格式 2：A\$ = INPUT\$(n)

功能：从键盘上读入 n 个字符，赋值给 A\$，且不在屏幕上回显。如果键盘上没有输入或输入的字符少于 n 个，则处于等待

状态。

4. 读数与置数语句

格式: READ <变量 1> [, <变量 2>...]

DATA <常数 1> [, <常数 2>...]

RESTORE [n]

说明:

① DATA 为非执行语句, 可放在程序的任何地方。

② DATA 中的常数与 READ 中的变量一般要一一对应, 类型要匹配。

③ DATA 中的数据个数多于 READ 中的变量数目时, 多余的数废弃不用; 读数时如果数据个数少于变量数, 会显示数据不足的错误。

④ DATA 中的字符串常数, 如果不出现逗号、冒号、前空格或尾随空格, 则可不必用双引号括起来, 否则必须用双引号括起来。

⑤ RESTORE n 语句用来定义数据指针的行号 n, 如缺少选择项 n, 则隐含指向第一个 DATA 语句的头一个数据。

5. 数据交换语句

格式: SWAP <变量 1, 变量 2>

功能: 将变量 1 与变量 2 的数据进行交换。

以上五种赋值方式各有特点: LET 语句中可以用表达式、函数等; INPUT 语句能实现人机对话; READ/DATA 语句适于对众多的变量一次赋值; INPUT\$(n) 输入的字母具有保密性。掌握它们的特点, 可以根据要求合理地使用它们。

三、运算及控制语句

运算功能多在 LET 语句中进行, 等号右边往往包含变量、函数、自定义函数及运算符。

1. 函数

函数	功能
ABS(x)	取 x 的绝对值
INT(x)	取 $\leq x$ 的最大整数
FIN(x)	将 x 截断成整数
SQR(x)	x 的平方根
EXP(x)	e 的 n 次幂
LOG(x)	x 的自然对数
SIN(x)	$\sin x$
COS(x)	$\cos x$
TAN(x)	$\operatorname{tg} x$
ATN(x)	$\operatorname{arc} \operatorname{tg} x$
SGN(x)	符号函数 $x > 0$ 时为 1, $x = 0$ 时为 0, $x < 1$ 时为 -1
TIME \$	机内时间(时:分:秒)
RND(n)	产生一个 0 到 1 之间的随机数
VAL(x \$)	将字符串变量 X \$ 变成数值
STR \$(X)	将数字 X 变成字符串
ASC(X \$)	取 X \$ 中第一个字符的 ASCII 码
CHR \$(n)	取 ASCII 码为 n 的字符
RIGHT \$(X \$, n)	取 X \$ 中最右边的 n 个字符
LEFT \$(X \$, n)	取 X \$ 中最左边的 n 个字符
MID \$(X \$, n, m)	取 X \$ 中从第 n 个字符开始的 m 个字符

关于 RND 的三点说明:

① 参数 n 为任选项。当 n 为 0 时仍取前一个随机数的值, 当 n 为非 0 值时可不选参数 n。

② 产生的随机数是“伪随机数”。例如要产生 10 个随机数, 每次运行总是这 10 个随机数。

③ 为了克服“伪随机数”的缺陷, 可使用“RANDOMIZE [n]”语句, 只要改变 n 的值, 各次产生的随机数就不再重复, 所以称这条语句为“重选随机数发生器”。

下面给出一种改变 n 值的方法:

```
10 'pml-2
20 T$=RIGHT$(TIME$,2)
30 N=VAL(T$)
40 RANDOMIZE N
50 PRINT RND, RND, RND
60 END
```

由于 N 是 0 到 59 之间的一个数, 在两次运行时重合的机率较小, 所以产生的随机数也就不同了。

2. 运算符

算术运算: +、-、*、/、\ (整除)、^、MOD (取余数)

关系运算: =、<、>、<> (不等于)、<=、>=

逻辑运算: NOT、AND、OR、XOR

3. 控制语句

* GOTO n——无条件转向语句

* IF <条件> THEN <语句 1> [ELSE <语句 2>]

说明:

① “条件”可以是关系表达式或逻辑表达式, 条件可以有多个。

② 语句 1(语句 2)可以是一个 BASIC 语句或用冒号":"隔开的语句序列,也可以是一个行号。

③ IF 语句可以嵌套使用,如果嵌套结构中有 ELSE,则以最近的未配对的 THEN 来与其相配。

* GOSUB/RETURN _____ 转子程序/返回

* ON <控制变量> GOTO/GOSUB <n1><,n2,...> _____ 开关语句。控制变量取 1 时,转 n1,取 2 转 n2,余类推。

* FOR <变量>=E1 TO E2 [STEP E3] ... NEXT _____ 步长型循环语句

* WHILE <条件>... WEND _____ 当条件型循环语句
说明:

① 当条件循环语句中的条件是一个关系表达式或逻辑表达式,它们的值都是一个逻辑值:真或假。真表示条件成立,假表示条件不成立。

② 条件成立时执行循环体,条件不成立时跳出当条件循环。

③ 在循环体中一般要有改变条件的语句,否则会出现死循环。但有时条件的改变蕴藏于条件之中。例如下段程序中

```
10 WHILE INKEY$=""  
20 ' 循环体  
30 WEND
```

这个条件的意思是:当无键盘输入时执行循环体,有任何键盘输入都会跳出当条件循环。

四、 输出显示语句

输出部分的作用是显示运算结果,下面介绍四种格式。

格式 1: PRINT ["说明项";]<变量 1> [<,/>;变量 2>...]

说明:

① 说明项为任选项,每个变量前都可以加说明项。变量可为

表达式。

② 变量后选逗号“,”为标准格式输出,选分号”;”为紧凑格式输出,由于数据前有符号位“±”,数据后有空格,所以在紧凑格式中数据之间也至少有一个空格。

③ PRINT 可以用问号“?”取代。

格式 2: PRINT TAB(n1); [“说明项 1”;] [变量 1;
[TAB(n2);...]

说明: n2 与 n1 之差应大于说明项 1 与变量 1 所占的格数之和。

格式 3: PRINT USING 格式字符串; 表达式 1 [, 表达式 2...]

说明:

① 格式字符串是用双引号括起来的约定符号。

例如:用“\ \”(两个反斜杠夹着 n 个空格)来规定其后的字符串型数据显示时都占用 n+2 个格。如果数据本身不足 n+2,就在右边填上空格;如果超出了 n+2;就截断右边超出的部分。

再如:用“###.###”来规定其后的数字型数据的显示格式。小数点左边“#”字的个数表示整数位,如果整数部分超出了规定的长度,仍显示超出的数字,但在数的前面显示一个“%”作为警告。小数点右边“#”的个数表示小数位,如果小数不足此长度自动补零,如果超出了此长度则进行舍入处理。

还有其他一些格式字符串,此处从略。

② 使用格式字符串后,数据之间不管是逗号还是分号,各数据之间都按格式字符串规定的长度紧紧相连。

格式 4: LOCATE (m,n): PRINT [“说明项”;] (变量)

说明:

① LOCATE 语句仅对屏幕显示起作用,对打印机输出不起

作用。

② m, n 为屏幕字符坐标, m 为行, n 为列。

五、结束语句

* RETURN _____ 子程序结束, 转回调用处。

* END _____ 主程序结束运行。

* CHAIN N\$ _____ 结束本程序运行, 转去执行 N\$ 程序。N\$ 是以双引号括起来的文件名。

例 1-3: 计算并打印开方表。

```
10 'pml-3: SQR. TABLE
15 CLS: ? TAB(20); "SQR. TABLE"
20 FOR I=0 TO 9: M=7 * (I+1)
25 ? TAB(M); I; " ";
30 NEXT: ?
35 FOR J=0 TO 9: ? J; "0"; \
40 FOR I=0 TO 9: M=7 * (I+1)-1
45 X=10 * J+I: X=SQR(X)
50 ? TAB(M); USING " #. # # "; X;
55 NEXT: ?
60 NEXT
65 END
```

运行结果如下:

SQR. TABLE

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
00)	0.00	1.00	1.14	1.73	2.00	2.24	2.45	2.65	2.83	3.00
10)	3.16	3.32	3.46	3.61	3.74	3.87	4.00	4.12	4.24	4.36
20)	4.47	4.58	4.69	4.80	4.90	5.00	5.10	5.20	5.29	5.39
30)	5.48	5.57	5.66	5.74	5.83	5.92	6.00	6.08	6.16	6.24